using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp11

{

interface IChain

{

IChain NextChain { get; set; }

void SetNextChain(IChain chain);

void Calculate(Numbers numbers);

}

class Numbers

{

public Numbers(int number1, int number2, string calculationWanted)

{

Number1 = number1;

Number2 = number2;

CalculationWanted = calculationWanted;

}

public int Number1 { get; set; }

public int Number2 { get; set; }

public string CalculationWanted { get; set; }

}

class AddChain : IChain

{

public IChain NextChain { get; set; }

public void Calculate(Numbers numbers)

{

if (numbers.CalculationWanted == "add")

{

Console.WriteLine(numbers.Number1 + numbers.Number2);

NextChain.Calculate(numbers);

}

else

{

NextChain.Calculate(numbers);

}

}

public void SetNextChain(IChain chain)

{

NextChain = chain;

}

}

class SubtChain : IChain

{

public IChain NextChain { get; set; }

public void Calculate(Numbers numbers)

{

if (numbers.CalculationWanted == "subt")

{

Console.WriteLine(numbers.Number1 - numbers.Number2);

}

else

{

NextChain.Calculate(numbers);

}

}

public void SetNextChain(IChain chain)

{

NextChain = chain;

}

}

class MultChain : IChain

{

public IChain NextChain { get; set; }

public void Calculate(Numbers numbers)

{

if (numbers.CalculationWanted == "mult")

{

Console.WriteLine(numbers.Number1 \* numbers.Number2);

}

else

{

NextChain.Calculate(numbers);

}

}

public void SetNextChain(IChain chain)

{

NextChain = chain;

}

}

class DivChain : IChain

{

public IChain NextChain { get; set; }

public void Calculate(Numbers numbers)

{

if (numbers.CalculationWanted == "div")

{

Console.WriteLine(numbers.Number1 / numbers.Number2);

}

else

{

Console.WriteLine("You can only add subt mult div");

}

}

public void SetNextChain(IChain chain)

{

NextChain = chain;

}

}

class Program

{

static void Main(string[] args)

{

Numbers numbers = new Numbers(3, 5, "mult");

IChain request1 = new AddChain();

IChain request2 = new SubtChain();

IChain request3 = new MultChain();

IChain request4 = new DivChain();

request1.SetNextChain(request2);

request2.SetNextChain(request3);

request3.SetNextChain(request4);

request2.Calculate(numbers);

}

}

}